Probabilistic Reasoning and Decision Making

INF581 Advanced Machine Learning and Autonomous Agents

Jesse Read

Last Updated: January 5, 2022

About this Course

We follow a thread through the large, general area of **constructing Intelligent Agents**.

(see Moodle for more information)

Outline

Introduction

What is AI?

> WIKIPEDIA: *AI is the study of "intelligent agents": any device that **perceives** its environment and **takes actions** that maximize its chance of success at **some goal***

We wish to build a function

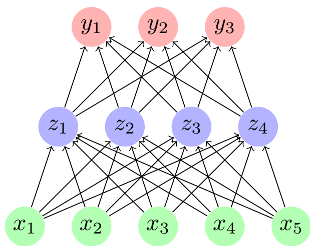
$$\boxed{f : \mathcal{P} \mapsto \mathcal{A}}$$

to map perception (e.g., of features, images, text, sensors, . . .) to action (e.g., decisions, predictions, apply-label, actuators) with some goal (i.e., performance metric) in mind.

The question is: How to build this function?

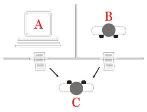
The existence of f

- Turing Machines (1936): Capable of simulating *any* computer algorithm
- Universal Approximation (1989): A simple neural network can approximate *any* continuous function f



So our function f can exist – we just need to learn it. How do we know when we 'get there'?

The Turing Test – a test for AI?



Already 'beaten' by AI [1] , but the agent . . .

> *dodges questions, he changes the subject, he makes vague answers, he repeats things back to you in ways that no normal human does in a cute attempt to show that he's listening,. . .*
>
> *[MGONZ] was combative, vulgar, and insulting. Any alarms going off in your head when testing MGONZ would be attributed to the mannerisms of an asshole, rather than a computer. Bypassing your brain's psychopath detector is a neat trick. . .*

- An emphasis on natural language
- and presumes that successfully faking 'human-ness' shows AI
- Can a machine think? *Can a submarine swim?* – E. Dijkstra
- Do we need AI to be like us (and who is *us*?)

Learning f from data?

There are major risks of encoding human bias, e.g., [2] (2016):

> *Microsoft's latest experiment in real-time machine learning, an AI-driven chat-bot called Tay, quickly turned to the dark side on Wednesday after the bot started posting racist and sexist messages on Twitter in response to questions from users. Among other things, Tay said the Holocaust never happened, and used offensive terms to describe a prominent female game developer.*

And [3] (2018):

> *Amazon has apparently abandoned an AI system aimed at automating its recruitment process . . . [It] penalised résumés that included the word 'women's', as in 'women's chess club captain' and marked down applicants who had attended women-only colleges . . .*

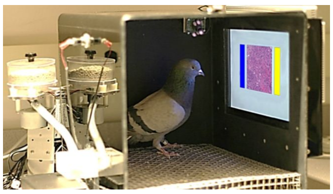
(more on this is Lecture 8)

Self-driving cars – a better test/showcase of AI?

A modest goal; getting from point A to B without crashing into anything is something a pigeon can do (in 3D!)

Not that we should underestimate pigeons [4] :

> ***Pigeons identify breast cancer 'as well as humans'***: *Pigeons, with training, did just as well as humans in a study testing their ability to distinguish cancerous from healthy breast tissue samples. [. . .] Likely no bigger than the tip of your index finger, the pigeon's brain nonetheless has impressive capabilities . . . Pigeons can distinguish identities and emotional expressions on human faces, letters of the alphabet, misshapen pharmaceutical capsules, and even paintings by Monet vs Picasso.*



https://www.scientificamerican.com/article/using-pigeons-to-diagnose-cancer/

Applications and Recent Progress

2004:

- Prof. of undergrad. AI course: *"I can teach you to play Go in a few minutes and you will be able to beat the state-of-the-art AI"*
- DARPA Grand Challenge: None of the autonomous vehicles finished; the best team completed 11.78 of the 240 km
- Computer vision largely unsolved, could not recognise faces with beards, etc. Focus on getting any kind of text associated with images on the web because we couldn't do much with pixels.
- Speech recognition had plateaued, needed heavy personalized training; Machine translation provided 'amusing' examples

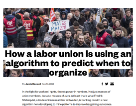
2022: Since a few years ago . . .

- Go world champion human has been beaten. Real-Time-Strategy games are a new frontier
- Autonomous vehicles can drive hundreds of kilometres through urban environments
- Computer vision and speech recognition penetrating the mass market (Siri, Google Translate, etc.)

Applications

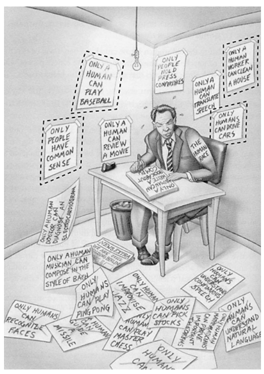
- Autonomous vehicles, robotics and navigation
- Banking / finance
- Smart grids, dynamic resource management
- Healthcare (diagnostics, drug discovery, . . .)
- Recommendation System
- Fraud prevention / anomaly detection / cybersecurity
- Agriculture
- Translation
- Gaming and games
- Marketing, face recognition
- Art and creativity
- Economy, social challenges, politics



The 'AI Effect'

According to the "AI Effect" AI problems are the ones we haven't solved yet or are in the process of solving.



Source: [5]

So, what are we doing in this course?

We address the task of trying to learn the function (agent)

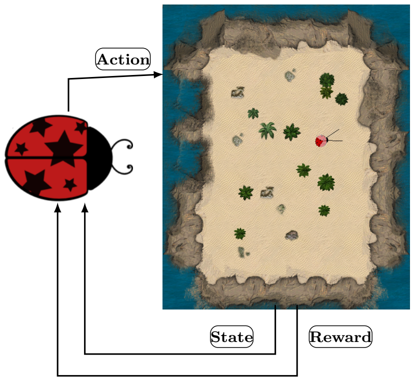
$$f : \mathcal{P} \mapsto \mathcal{A}$$

which maps perceptions (inputs) to actions (outputs), optimizing a suitable performance metric for the given task.

If successful we should get "good" performance: the agent is useful, impressive, or provides us with greater understanding (. . . ?)

We are interested in autonomous agents, and we them via advanced machine learning.

Agents



An agent maps perception to action in an environment. Which action? The one that maximizes its performance metric (reward).

Components of an Agent

To build

$$f : \mathcal{P} \mapsto \mathcal{A}$$

we need (in general):

- Perception ← We consider data and simulations (no robotics or sensors in this course)
- Knowledge and Representation ← Today (and Week 8).
- Reasoning and Decision Making ← Today
- Planning/Sequential decision making ← Much of the course
- Learning ← Week 2 and most of the course
- Acting ← We consider virtual actuators, e.g., which decision to take (no robotic actuators in this course)

Motivation: Reasoning, Knowledge and Planning

Should I cycle to work this morning? (Or take the train?)

- We cannot plan/decide if we cannot reason.
- Without reasoning, we would have to store infinite information (a lookup table is not viable!)
- We reason from knowledge (*e.g., weather, time, state of bike, state of self, strikes, . . .*).
- Thus we need to obtain knowledge (i.e., via perception, learning from experience),
- And we need to represent that knowledge for access and reasoning.

This is a difficult problem!

Another example: you message your personal-assistant agent:
Can you buy some beer for me for when I get home.

It needs to know . . .

What is beer (what does it look like, sound like, . . .)? What is it used for? How much does beer weigh? How much is 'some' beer? Where to obtain beer? When to obtain it? How to get there? How to transport beer? Does the beer need transporting? . . . What is home? Where is home? How to buy something? What is money? How much does beer cost? What happens if payment doesn't work? . . . Is this even a good idea to buy beer? Should I carry out this request? What are the consequences if I don't . . .

Bayesian Networks

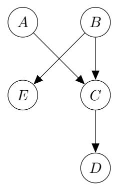
Probabilistic Graphical Models

Probabilistic graphical models (PGMs):

- A marriage between graphs and probabilistic models.
- Provide representation for knowledge, and mechanisms to reason from it
- Used throughout machine learning and beyond, especially when structured outputs are involved
- There are different types. We look at Bayesian Networks / Belief Networks. Another common variety is Markov Random Fields

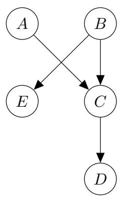
Bayesian Networks

- A directed acyclic graph (DAG), e.g.,



- Nodes are random variables, e.g., A, B, C, ...
- Edges represent conditional dependence between variables, e.g., $P(C|A,B)$ (note that A and B are the parents of C) – a distribution (we can represent with a probability table θ)
- N.B. Edge-relations are *not necessarily* causal – but can be
- The graph specifies/factorizes a joint probability distribution:

$$P(A,B,C,D,E) = P(A)P(B)P(C|A,B)P(D|C)P(E|B)$$



Generally: we have m variables, e.g.,

$$\boldsymbol{X} = \{X_1, \ldots, X_m\} = \{A, B, C, D, E\}$$

where each node X_j has a conditional probability distribution

$$P(X_j|\text{pa}(X_j))$$

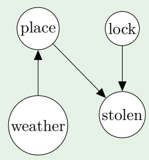
where $\text{pa}(X_j)$ are the parents of X_j, e.g., $\text{pa}(C) = \{A, B\}$.

The joint distribution, over the m variables/nodes, can be written as

$$P(\boldsymbol{X}) = \prod_{j=1}^{m} P(X_j|\text{pa}(X_j))$$

A Toy Example

The probability that my bike gets $\text{stolen} \in \{\text{yes}, \text{no}\}$, depends on what kind of $\text{lock} \in \{\text{D-lock}, \text{chain}\}$ I used, and the $\text{place} \in \{\text{outside}, \text{cage}\}$ where I left it. It does *not* depend *directly* on the $\text{weather} \in \{\text{sun}, \text{rain}\}$ (in this example).



So a Bayesian Network is a representation of knowledge; coming from either a domain expert, or experience/data (we must learn it).

A main task in PGMs is reasoning (inference) – from our knowledge.

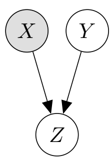
Probabilistic Reasoning (Inference)

We formulate a query, given some observed event/evidence, e.g.,

$$P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x})$$

i.e., with

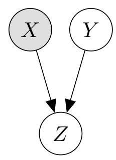
1. evidence, $\boldsymbol{x}$ N.B. Observed-evidence nodes are shaded
2. query, $\boldsymbol{y}$
3. knowledge in the form of the graph



Other variables that we are *not* interested in (wrt the query), e.g., $\boldsymbol{Z}$, are called hidden, latent, or 'nuisance' variables.

Marginalizing Out (Sum Rule; Law of Total Probability)

We marginalize out (sum out) the hidden variables.

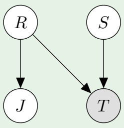


For example, with knowledge P (as defined by the PGM above), and observation x, we want to reason about y:

$$\begin{aligned}
P(Y = y, X = x) &= P(y, x, Z) & \triangleright \text{ Plug in query and observation} \\
&= P(Z|x,y)P(x)P(y) & \triangleright \text{ Factorize} \\
&= \sum_z P(z|x,y)P(x)P(y) & \triangleright \text{ Marginalize out } Z \\
&= P(x)P(y)\sum_z P(z|x,y) & \triangleright \text{ Push sum right}
\end{aligned}$$

Example[a]: Tracey's Grass

[a] from D. Barber, *Bayesian Reasoning and Machine Learning*



Tracey observes wet grass ($T = 1$). Was the Sprinkler on ($S = 1$)?

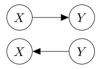
Relevant knowledge:

$P(R = 1) = 0.2$	(sometimes it *Rains*)
$P(S = 1) = 0.1$	(sometimes *Sprinkler* is on)
$P(T = 1 \mid R = 1, S = 0) = 1$	(*Rain* always wets grass)
$P(T = 1 \mid R = 0, S = 0) = 0$	(only *Rain*/*Sprinkler* cause wet)
$P(T = 1 \mid R = 0, S = 1) = 0.9$	(*Sprinkler* usually wets grass)
$P(T = 1 \mid R = 1, S = 1) = 1$	(... and with rain, always)

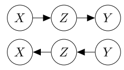
($J = 1 \Leftrightarrow$ Jack's grass is wet, but we don't care in this example)

Dependence: When are X and Y dependent?

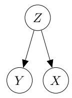
- Direct connection



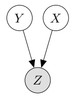
- Indirect connection (cascade)



- Common parent ('cause')



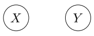
- Common child (explaining away); N.B. Z is *observed* as evidence!



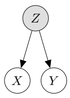
N.B. It does not matter if X or Y is observed/shaded or not

Independence: When are X and Y independent?

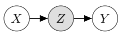
- Not connected (marginal independence)



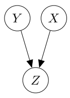
- Z is common evidence (conditional independence)



- Z blocks path between X and Y (conditional independence)



- Z is a collider node (N.B. Z is *not* observed in this case)



N.B. It does not matter if X or Y are observed/shaded or not

Marginal vs Conditional Independence

When X is (marginally) independent of Y we write:

$$X \perp\!\!\!\perp Y$$

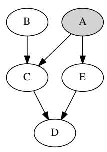
When X is conditionally independent of Y given Z we write:

$$X \perp\!\!\!\perp Y | Z$$

Recall that dependence is symmetric:

$$X \perp\!\!\!\perp Y \Leftrightarrow Y \perp\!\!\!\perp X, \quad \text{and} \quad X \perp\!\!\!\perp Y|Z \Leftrightarrow Y \perp\!\!\!\perp X|Z$$

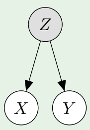
Example: Is $C \perp\!\!\!\perp E|A$?



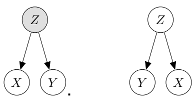
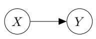
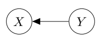
Answer: Yes, influence is blocked above by 'common evidence', and blocked below by 'collider'.

Speeding

€Fine $\perp\!\!\!\perp$ Type of Car | Speed (let Z be observed speed/evidence)



This does not imply that X and Y would be independent of each other *without* evidence Z! Note, if we have to marginalize out Z:

 $\Rightarrow$  or 

$$P(X, Y|Z = z) = P(X|Z = z) \cdot P(Y|Z = z) \quad \triangleright\ X \perp\!\!\!\perp Y|Z$$

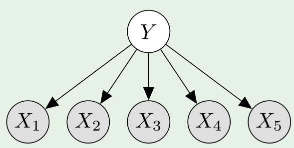
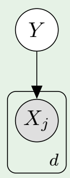
$$P(X, Y) = \sum_{z \in \mathcal{Z}} P(X|Z = z) \cdot P(Y|Z = z) \cdot P(Z = z) \quad \triangleright\ X \not\perp\!\!\!\perp Y$$

(Car-insurance companies know this!)

(Amazon's 'sexist AI' for recruiting: they marginalized out gender!)

Naive Bayes as a Bayesian Network

(An independence assumption in action)

 $\equiv$  (equiv. plate notation)

$$P(Y = y, \boldsymbol{X} = \boldsymbol{x}) = P(Y = y) \prod_{j=1}^{d} P(X_j = x_j | Y = y)$$

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(y|\boldsymbol{x}) = \underset{y}{\operatorname{argmax}}\, P(y, \boldsymbol{x})$$

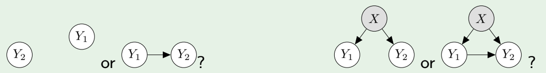
Multi-label Classification (Independence vs Common Evidence)

We have one input X and two class labels Y_1, Y_2, each $\in \{0, 1\}$

X	Y_1	Y_2	$P(X, Y_1, Y_2)$
0	0	0	0.25
0	0	1	0
0	1	0	0
0	1	1	0.25
1	0	0	0
1	0	1	0.25
1	1	0	0.25
1	1	1	0

Example from Dembczynski et al., *On Label Dependence in Multi-Label Classification*

Are Y_1, Y_2 marginally/conditionally independent (given $x = 1$)?

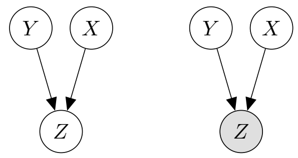


$$P(Y_1, Y_2) = P(Y_1)P(Y_2)$$

$$\sum_{x\in\{0,1\}} P(Y_1|x)P(Y_2|Y_1, x) = \sum_{x\in\{0,1\}} P(Y_1|x) \sum_{x\in\{0,1\}} P(Y_2|x) \quad \triangleright \text{ marginal dep.?}$$

Explaining Away

Recall: Two variables become dependent if a common child (have a common effect) is observed



If Z is *not* observed (left),

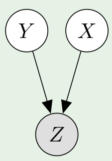
$$\begin{aligned} P(X, Y) &= \sum_z P(X, Y, Z = z) \\ &= P(X)P(Y) \underbrace{\sum_z P(Z = z|X, Y)}_{1} \\ &= P(X)P(Y) \quad \triangleright \text{ independence} \end{aligned}$$

but if the common child is observed (right),

$$P(X, Y|Z = z) = \frac{P(X, Y, Z = z)}{\ldots} \quad \triangleright \text{ cond. prob.}$$

Explaining Away: Missing Bike Example

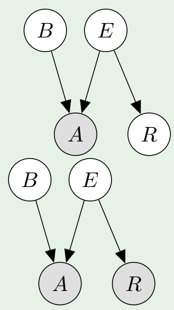
My friend borrowing my bike ($X = 1$; suppose he has a set of keys) and a thief stealing it ($Y = 1$) occur independently of each other. But if my bike is missing ($Z = 1$), then X and Y are now dependent: if $P(X = 1)$ is high (he told me he would), then $P(Y = 1)$ becomes low, i.e., my friend 'explains away' much possibility of theft.



Explaining away: Earthquake Example[a]

[a] from D. Barber, *Bayesian Reasoning and Machine Learning*

The house alarm is sounding ($A = 1$); was it a burgler ($B = 1$), or was it triggered by an earthquake ($E = 1$)? There is a report on the radio ($R = 1$) about an earthquake.



Alarm = 1	Burglar	Earthquake
0.9999	1	1
0.99	1	0
0.99	0	1
0.0001	0	0

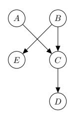
Radio = 1	Earthquake
1	1
0	0

$P(B = 1) = 0.01$; $p(E = 1) = 0.000001$.

$$p(B = 1|A = 1) = \frac{\sum_{E,R} p(A = 1|B = 1, E)p(B = 1)p(E)p(R|E)}{\ldots}$$

Bayesian Networks: Summary

- A DAG specifies conditional dependence relations
- Each node is a conditional distribution, parametrized by θ
- A Bayesian Network is a representation of knowledge
- We can do reasoning/inference on it across multiple variables by marginalizing out the variables we don't need.
- We should get θ from an expert or [mainly in this course] by learning from data and/or experience.



Links different areas of mathematics: probability and graph theory. Used throughout machine learning and AI, including in deep learning, and in many modern challenging applications domains. Provides us with important components of intelligent agents: knowledge and reasoning.

Decision Theory

Decision Theory: From Beliefs to Actions

Decision Theory = Probability Theory + Utility Theory

We have quantified our knowledge (beliefs) probabilistically via a Bayesian network (aka belief network).

Based on these beliefs, and perception/observed evidence, we can make decisions ...

- What labels should I apply to an image?
- Should I speed?
- Should I accept a job offer?
- ...

We require an error function (loss, lower is better) or payoff function (utility, higher is better).

Toy Example

Suppose observation/state $S \in \{s_1 \equiv \text{Rain}, s_2 \equiv \neg\text{Rain}\}$,
action/decision $A \in \{a_1 \equiv \text{Leave umbrella}, a_2 \equiv \text{Take umbrella}\}$
payoff/reward function $u(\text{decision}, \text{state})$:

	s_1 (Rain)	s_2 ($\neg$Rain)
a_1 (Leave umbrella)	-40	60
a_2 (Take umbrella)	10	-20

Which decision should I take (take my umbrella or leave it)?

- Optimist agent: select the max. reward for each decision $(60, 10)$, then select the (arg)max. of those (i.e., action a_1)
- Pessimist agent: select the min. reward for each decision $(-40, -20)$, and then the (arg)max. of those (i.e., action a_2)

What should a rational agent do? (one who takes the best action)

Answer: Maximize expected reward (i.e., minimize expected loss)
If it assumes $P(S = s_1) = P(S = s_2) = 0.5$: action a_1, $\mathbb{E}[J(a_1)] = 10$, $\mathbb{E}[J(a_2)] = -5$.

Decision Theory in Machine Learning

In machine learning, we wish to assign/predict label(s)

$$\widehat{\boldsymbol{y}}^* = h(\boldsymbol{x})$$

i.e., the best hypothesis, $\widehat{\boldsymbol{y}}^* \in \mathcal{Y}$, given observation $\boldsymbol{x}$, according to our loss function ℓ.

The optimal strategy is to minimize conditional expected loss (also known as risk),

$$\begin{aligned}\widehat{\boldsymbol{y}}^* &= \operatorname*{argmin}_{\widehat{\boldsymbol{y}} \in \mathcal{Y}} \mathbb{E}_{\boldsymbol{Y} \sim P(\boldsymbol{Y}|\boldsymbol{x})}[\ell(\widehat{\boldsymbol{y}}, \boldsymbol{Y})|\boldsymbol{x}] \\ &= \operatorname*{argmin}_{\widehat{\boldsymbol{y}} \in \mathcal{Y}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \ell(\widehat{\boldsymbol{y}}, \boldsymbol{y}) P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{x})\end{aligned}$$

Remark: This requires P (or an estimation thereof, e.g., from a Bayesian network).

Example: MAP Estimate

In the case of 0/1 loss ($\ell(y, \widehat{y}) = 1$ if $y \neq \widehat{y}$, else 0)[1]:

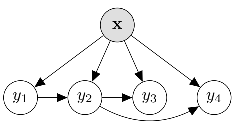
$$\widehat{\boldsymbol{y}}^* = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y}|\boldsymbol{x})$$

This is the maximum a posteriori (MAP) estimate.

[1] If $\boldsymbol{y} \in \mathcal{Y}^L$ then $\ell(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = 0$ if $\sum_{j=1}^{L} \ell(y_j, \widehat{y}_j) = 0$ (and 1 otherwise)

Example: Multi-label Classification

Observe $\boldsymbol{x}$, want to minimize 0/1 loss of decision $\widehat{\boldsymbol{y}} = [\widehat{y}_1, \ldots, \widehat{y}_4]$, where $y_j \in \{0, 1\}$, given knowledge



Minimizing the 0/1 loss,

$$\widehat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \{0,1\}^4} P(\boldsymbol{y}|\boldsymbol{x})$$

i.e., $\boldsymbol{y} \in \{[0,0,0,0], [0,0,0,1], \ldots, [1,1,1,1]\}$.

This provides our decision function

$$\widehat{\boldsymbol{y}} = h(\boldsymbol{x})$$

Rational Agents

A rational agent (acts in a way to minimize loss) considers

1. Actions $\mathcal{A} = \{a_1, \ldots, a_J\}$
2. States $\mathcal{S} = \{s_1, \ldots, s_K\}$
3. Loss $\ell(a|s)$ of taking action $a \in \mathcal{A}$ under true state $s \in \mathcal{S}$.

and will model uncertainty[2] (e.g., which state am I in?) via $\mathbb{E}$, built from P the knowledge of the agent, under some evidence $\boldsymbol{x}$.

The expected loss (risk) of taking action a under observation $\boldsymbol{x}$, is

$$R(a|\boldsymbol{x}) = \mathbb{E}_S[\ell] = \underbrace{\sum_{s \in \mathcal{S}} \ell(a|s) \cdot P(s|\boldsymbol{x})}_{\text{Expected Loss aka Risk}} \tag{1}$$

and thus optimal action is

$$a^* = \operatorname*{argmin}_{a \in \mathcal{A}} R(a|\boldsymbol{x}) \qquad \triangleright \text{ minimizing Eq. (1)} \tag{2}$$

[2] Unless in a deterministic world

Equivalent to maximizing payoff fn/utility. Therefore, equivalently, a rational agent *maximizes*

Expected Utility

$$U(a|\boldsymbol{x}) = \sum_{s \in \mathcal{S}} u(a|s) P(s|\boldsymbol{x})$$

of an action a under observation $\boldsymbol{x}$.

with utility $u(a|s)$ of action a from state s. So,

$$a^* = \underset{a \in \mathcal{A}}{\operatorname{argmin}}\, R(a|\boldsymbol{x}) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, U(a|\boldsymbol{x})$$

Different agents may have different utility functions, even when output is the same item, i.e., model *loss* differently wrt, e.g., life, accepting a position, making a wrong classification, etc.

Money Matters

- Option A: Recieve €1; or
- Option B: Recieve €100 with probability 0.0125?

Payoff function u in the literal sense (more money is better).

$$U(A) = \mathbb{E}[u(A)] = 1, \text{ and } U(B) = \mathbb{E}[u(B)] = 1.25$$

and, again, through an argmax (i.e., Eq. (2)),

$$\hat{y} = \underset{y \in \{A,B\}}{\operatorname{argmax}}\, \mathbb{E}[u(y)] = B$$

So we should choose option B...?

Utility Functions: Risk-Prone vs Conservative Agents

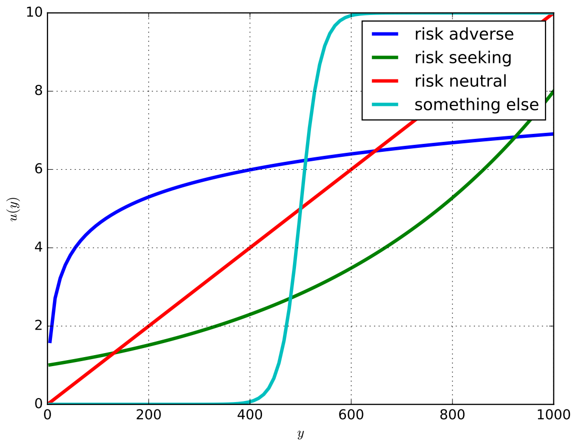
The *utility* of the payoff differs from the *size* of the payoff.

The utility function is agent-specific (as a loss function is task-specific):

- A risk-prone agent will tend to gamble higher stakes,
- A conservative (risk-adverse) agent will not (=*most people!*, hence insurance companies).
- A risk-neutral agent only cares about the size of payoff directly, i.e., $u(y) = y$

Regarding wealth: This depends on age, existing wealth, relative wealth to other agents, ability to acquire wealth through other means, ...

Remark: Other agents may not have the same loss function or the same model of the environment/beliefs!



Money Matters II – The Utility of Money

If the agent has a utility function $u(y) = \sqrt{y}$ to outcome y (i.e., is risk adverse), then – returning to earlier example – if

- Option A: Recieve €1; or
- Option B: Recieve €100 with probability 0.0125?

Utility:

u(A)	$\sqrt{1} = 1$
u(B)	$\sqrt{100} = 10$

$$\begin{aligned} U(B) &= P(y = 100) \cdot u(100) \\ &= 0.0125 \cdot 10 = \mathbf{0.125} \\ U(A) &= P(y = 1) \cdot u(1) \\ &= 1 \cdot 1 = \mathbf{1.0} \end{aligned}$$

then option A is much more attractive (for this agent).

Loss of Life

> *Program A: If adopted, exactly 400 out of 600 patients will die. Program B: If adopted, the probability that nobody will die is 1/3, while the probability that all 600 will die is 2/3.*

The expected loss of life is equivalent, $\mathbb{E}[\ell(A)] = \mathbb{E}[\ell(B)]$. Therefore the utility is also equivalent, $U(A) = U(B)$. Yet *only 13 percent of the doctors*[a] in this group chose to administer program A!

[a] According to [6]

Doctors are not rational agents???

Irrational Agents

Decision theory describes how a rational agent should act. But humans are not rational agents! Rational agent $\neq$ intelligent agent.

- "Human irrationality" (vs rational agents)
- Phrasing questions in different ways
- Herd mentality
- Idealogy
- Multiple agents (game theory)
- Bias
- Ego ($>$ 50% of people believe they are more intelligent than average, better-than-average drivers, etc.)

Example 1, Example 2

Our brains did not evolve to process utility tables. Recall: We don't always need to emulate humans (but we should keep this in mind).

Decision Theory: Summary

- States and observations
- Actions/decisions
- Loss/cost function or payoff/utility/reward function
- The design/choice of these is important!
- Knowledge representation P
- Modeling uncertainty via expectations $\mathbb{E}$ with P
- Optimal action (as taken by rational agents): the one which minimizes expected loss or –equivalently– maximizes expected payoff

i.e., we have completed a pipeline for an intelligent agent, f, from perception (observations of states) to actions (decisions).

Summary

Summary

We are building towards 'intelligent agents':

- Perception of its environment (e.g., *inputs* $\boldsymbol{x}$)
- Knowledge/representation (e.g., Bayesian Networks, P)
- Reasoning and making decisions under uncertainty (as $\mathbb{E}$)
- Actions (e.g., *outputs* $\boldsymbol{y}$)

i.e., a complete framework f from perception to action.

Focus of this lecture: Introduction probabilistic graphical models and probabilistic reasoning/inference.

What's missing (recall, Slide 15)/coming up in the next lectures: Learning (multi-output, structured output, reinforcement learning), and Planning (sequential decision making).

Probabilistic Reasoning and Decision Making

INF581 Advanced Machine Learning and Autonomous Agents

Jesse Read

ÉCOLE POLYTECHNIQUE
IP PARIS